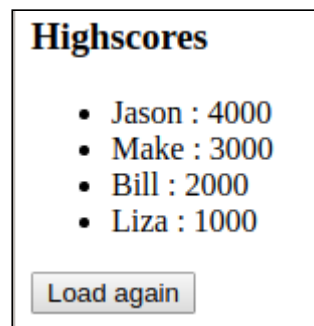
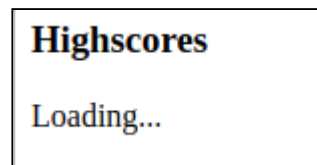


Harjoitukset 9

- Harjoitustehtävät palautetaan [ScoreTronicilla](#). Tarjolla on myös [ScoreTronicin käyttöohjeet](#)
- Harjoitukset 9 tulee palauttaa ryhmittäin seuraavasti
 - TTV18S2: 13.11.2019 klo 12.30 mennessä
 - TTV18S3: 12.11.2019 klo 11.30 mennessä
 - TTV18SM: 4.12.2019 klo 23.59 mennessä

Tehtävä 1 [4p]

Toteuteta highscore-tulosten esittäminen React-komponentin avulla. Kun tuloksia ladataan (3-4sec), käyttäjälle esitetään `Loading...`-tekstiä- Kun lataus on valmis, tulokset esitetään kuvan mukaisesti UL-listana.



highscore-data voidaan "kovakoodata" JavaScriptin avulla:

```
1 let highscoreData = [  
2   { id: 1, name: 'Jason', score: 4000 },  
3   { id: 2, name: 'Make', score: 3000 },  
4   { id: 3, name: 'Bill', score: 2000 },  
5   { id: 4, name: 'Liza', score: 1000 }  
6 ];
```

Komponentin tila voidaan alustaa:

```
1      constructor(props) {  
2          super(props);  
3          this.state = {highscores: [], loaded: false};  
4          // Mahdollisesti muuta toiminnallisuutta  
5      }
```

Latauksen "hitaus" saadaan aikaan käyttämällä `setTimeout`-funktiota esim. seuraavasti:

```
1      getHighscores () {  
2          setTimeout(() => this.setState({highscores: highscoreD  
3      }
```

Palautus

Tehtävä palautetaan osoittamalla URL toimivaan ohjelmaan

Tehtävä 2 [4p]

Toteuta edelleinen tehtävä 1 siten, että highscore-data on tallennettu erilliseen JSON-tiedostoon ja se ladataan sovellukseen AJAX-toiminnallisuutta käyttäen. Tässä tehtävässä datan lataamisen hidastusta ei tarvitse erikseen keinotekoisesti luoda.

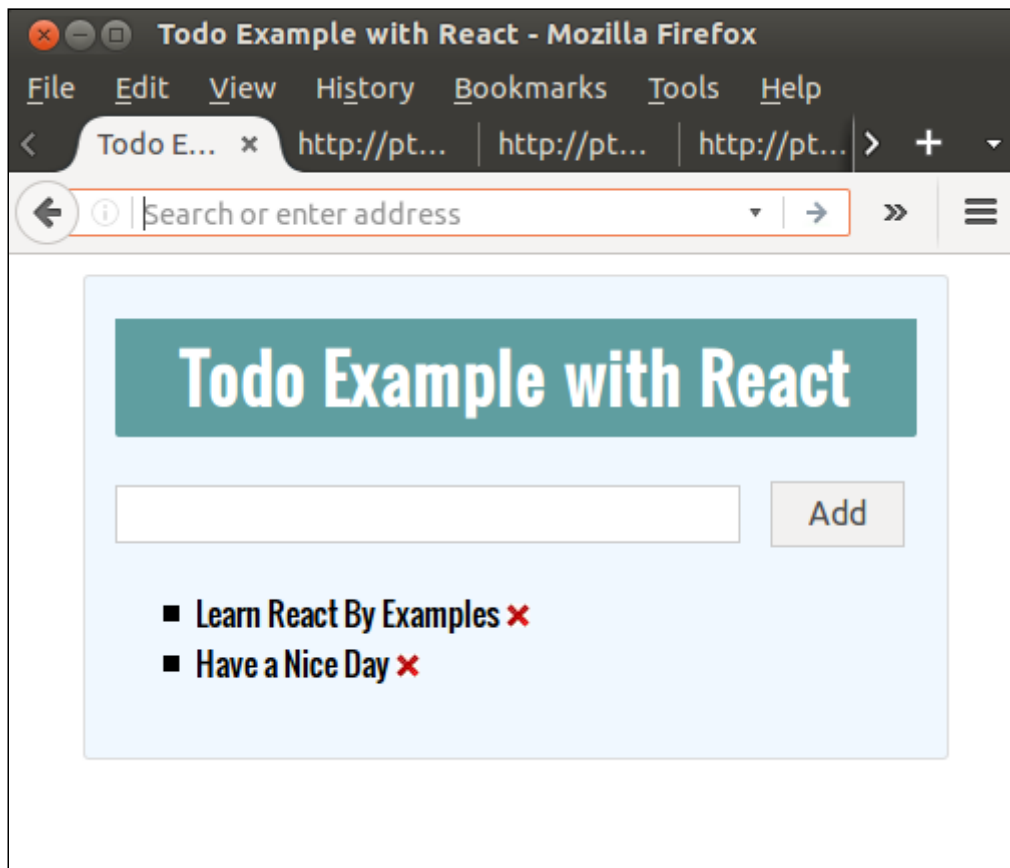
Palautus

Tehtävä palautetaan osoittamalla URL toimivaan ohjelmaan

Tehtävä 3 [4p]

Kirjoita React-kirjaston avulla ToDo-sovellus, joka säilyttää kulloisenkin tehtävälistan App-komponentin `this.state.items`-tilamuuttujataulukossa. Tehtäviä tulee voida lisätä listaan lomakkeen avulla ja tehtäviä tulee voida myös poistaa tehtävälistasta (punainen rasti). Ohjelmakoodin tulee pohjautua ja käyttää ohessa annetuissa tiedostoissa käytettyä komponentteja ja niiden välisiä suhteita.

Ohjelmakoodin pohja: [h8t03-init](#)

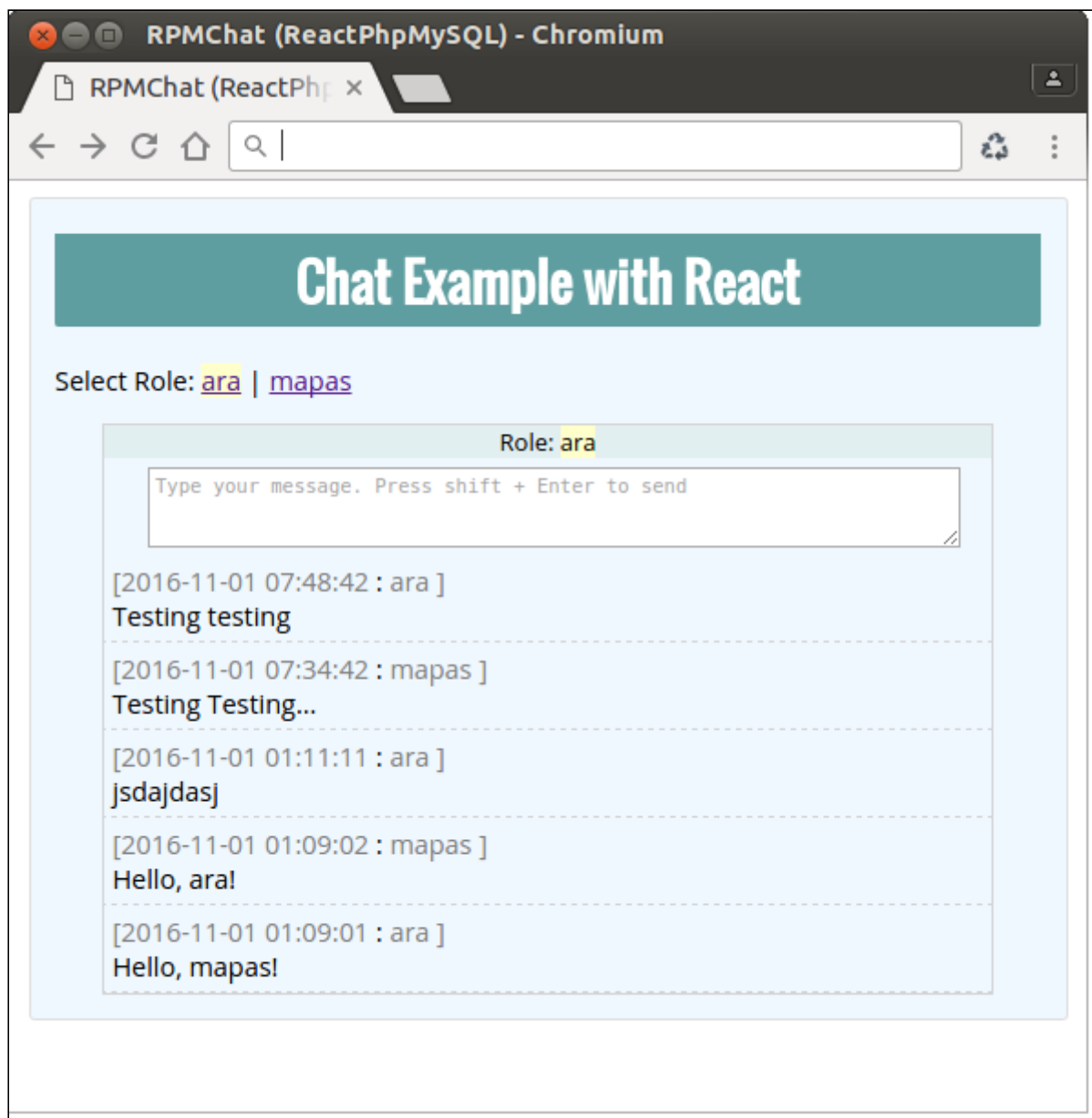


Palautus

Tehtävä palautetaan osoittamalla URL toimivaan ohjelmaan

Tehtävä 4 [4p]

Asenna ja ota käyttöön luvun <http://netisto.fi/oppaat/js/11.html> mukainen OnLineChat-sovellus. Muuta käyttöliittymän päänäkymää vähintään sen verran, että muutoksen huomaa ponnistelematta. Lisää myös pari uutta sinun lähettämiksesi tunnistettavissa olevaa viestiä keskusteluun.



Palautus

Tehtävä palautetaan osoittamalla URL toimivaan ohjelmaan **SEKÄ** kuvaruutukaappaukseen käyttöliittymästäsi siten että myös lisäämäsi viestit näkyvät chatissa

Tehtävä 5 [4p]

Tee **mikä tahansa** tämän opintojakson aiempi Reactiin liittyvä tehtävä käyttämällä node.js:ää ja npm:ää. Asenna ko. ohjelmistot tarvittaessa järjestelmääsi. Alla pari ohjetta ko. ympäristön käyttöönottoon ja sovelluksen luomiseksi.

- <https://www.robinwieruch.de/react-js-windows-setup/>

- <https://github.com/facebook/create-react-app>

Totea, että voit ajaa tekemääsi ohjelmaa paikallisesti työasemassasi ja luo sen jälkeen sovelluksestasi tuotantoversio (esim. `npm run build`). Kopioi tuotantoversio julkiselle web-palvelimelle esim. `student.labranet.jamk.fi` ja testaa myös sen toiminta.

Selvitä lisäksi montako tiedostoa sovelluksesi käytössä on kehitysympäristössä ja mikä niiden yhteiskoko on. Selvitä samoin montako tiedostoa sovelluksesi käytössä on tuotantoversiossa ja mikä niiden yhteiskoko on. Kirjoita nämä tiedot ytimekkäästi selkokielellä tiedostoon `reactapp-dev-vs-build.txt`

Palautus

Tehtävä palautetaan osoittamalla URL toimivaan ohjelmaan **SEKÄ** tiedostoon `reactapp-dev-vs-build.txt`

Jätetty tarkoituksella tyhjäksi